# Merging RACF Databases

BY ROBERT S. HANSEL & ROBERT L. WHITTLE

Mainframe installations managing multiple RACF databases serving separate z/OS images may find significant advantages in merging them to form a single shared database. Chief among these advantages is establishing consistent security across multiple system images. Another is reducing and simplifying security administration by eliminating multiple executions of the same commands. Furthermore, when a shared database is implemented in a Sysplex, the installation can activate several valuable performance tuning features.

The creation and implementation of a common shared database can be quite complex depending on the extent to which profiles overlap and implementation philosophies differ between the existing separate databases. For example, we've encountered situations where the JESSPOOL profiles are designed to be sysout owner-centric (second qualifier in the resource name) on one database and job name-centric (third qualifier) on another. These two profile design approaches don't mesh well and remediation is required. The entire merge effort must be performed with great care to ensure current production work isn't disrupted on any of the affected systems. This article outlines the process for merging databases and provides tips and strategies for successfully completing the effort.

## Preparation

First, consider the ultimate objective

of the project. Are you:

- Building a merged database to be shared by several system images?
- Merging z/OS images along with RACF to form a single z/OS image?
- Migrating certain applications and their RACF profiles from one image to another?
- Building a merged database that will be copied to multiple systems and maintained using RACF Remote Sharing Facility (RRSF)?

Your objective will determine the level of coordination and staff participation necessary and dictate the sequence of tasks to be completed.

Now you need to obtain management commitment. This may be the most important step because senior management must understand that significant staff resources may be needed to prepare the databases for merger and that multiple Initial Program Loads (IPLs) may be required to implement changes. This may force management to forego certain Service-Level Agreement (SLA) commitments. Management directives also may be needed to suspend non-essential RACF administrative work and unrelated RACF projects, allowing staff to focus on this effort and avoid the added complexity caused by a constantly changing RACF environment.

Next, assemble a project team. The composition of this team transcends RACF administration. The z/OS systems programmers often need to make changes to configuration options, tables, and exits in z/OS, RACF, and other system products. Storage administration may get involved in dedicating DASD volumes to the new database. Application developers, resource owners, and users will need to be consulted about proposed profile changes. Everyone should be expected to help with testing. Due to the level of effort, risk, and complexity associated with this type of project, many installations find it beneficial to engage experienced consultants to provide guidance and assistance.

The next step is to develop a plan with detailed tasks, timeframes, and staff assignments. You'll need to perform an initial high-level comparison of the system images and RACF databases to identify issues to address during the merge project. Review both RACF and system exits, RACF tables, Class Descriptor Table (CDT) class entries, STARTED class entries, SETROPTS options, OPERATIONS assignments, and active resource classes along with their profiles, especially FACILITY class profiles, to identify significant differences to resolve. Also identify data set High Level Qualifiers (HLQs) common to all systems because their profiles and access lists will need to be integrated. The primary objective of this review is to give you a feel for the level of effort you'll need to devote to each of the major synchronization tasks.

Now you can determine the methodology and tools you plan to use in merging the contents of the databases. You might simply copy certain profiles from one database to the others over an extended time period using RACF commands, or you could use a utility to combine databases all at once. You may opt for some combination of the two. The methodology you select will determine the tools you will employ. These will include some combination of RACF utilities, free IBM RACF software tools, and commercial software products for creating commands to copy profiles, comparing databases profiles, comparing system and RACF configuration options and tables, and combining databases. (IBM's free RACF utilities are available via their RACF Downloads Webpage at www-03.ibm.com/servers/eserver/zseries/zos/racf/goodies.html.)

- Use the IRRRID00 Remove ID utility to identify and eliminate obsolete profile references
- Deactivate unused resource classes (be mindful of shared POSIT values as you might turn off a needed class)
- Eliminate obsolete users, groups, data set, and general resource profiles
- Remove unnecessary Global Access Table entries or ones that undermine underlying profiles
- Delete STARTED class and ICHRIN03 entries associated with obsolete started tasks.

Figure 1: Pre-Merge RACF Database Cleanup Tasks

### Database Cleanup

Database cleanup is an important first step. You want to get rid of the dead wood that can waste valuable analysis time. Focus on resource classes and profiles the merging databases have in common. For instance, if two databases contain profiles with the HLQ SYS2, determine if any of the profiles is obsolete and can be deleted before proceeding with synchronization and merge tasks. Figure 1 shows a list of cleanup actions to perform.

As part of cleanup, verify the integrity of the RACF databases and fix any errors. RACF utilities that will alert you to various types of database errors include:

- IRRUT200: Database Verification
- IRRUT400: Database Split/Merge/Extend
- IRRDBU00: Database Unload.

IRRDBU00 is especially effective for this task; it must look at almost every field in every profile.

### Synchronization

Synchronization is the process where you equalize and integrate the configuration, options, and profiles in separate RACF implementations and databases, paying special attention to areas where they overlap and potentially conflict. For example, if the tape data set protection option, TAPEDSN, is active on only one database, the option will either need to be activated on all databases or deactivated on the one to make them all identical. Likewise, if a profile exists on all databases (e.g., SYS1.**), its access list and attributes such as Universal Access (UACC) must be made the same on all. The goal of synchronization is to ensure all the z/OS images that will ultimately share one database can function properly with the same RACF settings. Often, it's helpful to designate one database as the target into which profiles from the other databases will be copied to build the final merged database. Whenever practical, synchronization changes should be completed on all systems before the final merge and implementation of the shared database.

The synchronization process involves a careful and detailed analysis, looking at all systems on an option-by-option, class-by-class and profile-by-profile basis, including individual profile fields where necessary. Two questions this analysis should attempt to answer are:

# Turn your Mainframe into your Enterprise Security Server

SECURITY TRAINING
SECURITY & ACCESS MANAGEMENT SOFTWARE
AUDIT & REMEDIATION ASSESSMENTS
YOUR ENTERPRISE
IDENTITY MANAGEMENT SOFTWARE
VANGUARD SECURITY CONFERENCE
AUDITING & COMPLIANCE SOFTWARE

Vanguard's complete line of security and auditing software, and training, conference, and audit and remediation assessment services helps you solve your enterprise security and audit challenges.

- The Vanguard Security and Access Management Solution enables your IT security personnel to work faster and more accurately.
- The Vanguard Auditing and Compliance Solution streamlines processes and helps you comply with regulations such as SOX, HIPAA and PCI.
- The Vanguard Identity Management Solution provides integrated authentication capabilities for your increasing number of enterprise systems and devices, and strong authentication enterprise wide without the need for expensive tokens.

Vanguard's enterprise security and audit solutions and services provide everything you need to turn your mainframe into your enterprise security server.

**Contact Vanguard at**
**www.go2vanguard.com**

**VANGUARD**
**INTEGRITY PROFESSIONALS, INC.**
ENTERPRISE SECURITY SOFTWARE

- Will an exit, table entry, option, or resource class that's active on only one system image adversely affect the operation of the others? For example, the JESJOBS controls job submission. It has a default return code of 8, meaning that no access is allowed if there's no covering profile. If the class is active on one system and not the others, suddenly activating it on the others could seriously impair production batch processing.
- Will a profile and its attributes or access list on one database adversely affect access on the other databases by either granting or denying access? Suppose user SJONES has OPERATIONS authority on one database and not the others. If SJONES isn't configured with OPERATIONS in the merged database, the user could experience a denial of access on the systems previously served by the one database. Conversely, if SJONES is given OPERATIONS in the merged, shared database, the user may gain inappropriate access authority on the other systems.

Areas to address in the synchronization process include:

- **System and RACF exits and interfaces:** To ensure consistency of functionality, all systems sharing a RACF database should implement the same exit code, security-related z/OS configuration options (e.g., Program Property Table [PPT] NOPASS entries), and RACF interfaces. This encompasses more than just RACF exits. System exits and software that interact with RACF will need to be synchronized, too. While it's conceivable each z/OS image could have its own unique set of exits and RACF interfaces, this often leads to confusion that can result in security exposures. Try to implement the same code on all systems.
- **RACF tables:** Pay particular attention to the ICHNCV00 naming conventions table, ICHRDSNT data set name table, ICHRRNG database range table, ICHRRCDE CDT as well as its companion CDT class, and the Global Access Table. ICHNCV00 rearranges data set names before checking profiles and, if not synchronized, could result in unexpected profiles protecting certain data sets. If the database is split into multiple data sets, all images will need the same ICHRDSNT and ICHRRNG configuration. Focus most of your attention on the ICHRRCDE and the CDT class. Identify and resolve inconsistencies in resource class definitions such as profile format, default return codes, OPERATIONS authority, and especially POSIT values.

```
 RACF Database A Profiles       -->
      <--     RACF Database B Profiles

  SYS2.BMC.PLIB%%.*  -->
      <--     SYS2.BMC.PLIB*.**
  SYS2.BMC.PRODMAST  -->
      <--     SYS2.BMC.P*.**
  SYS2.*.**     -->
      <--     SYS2.**
```

Figure 2: Integrating Data Set Profiles

- **SETROPTS options:** There are two major areas to address. First is resource class status. Prior to the merge, set every class in all databases to the same level of activation, including CLASSACT, GENERIC, GENCMD, AUDIT, STATISTICS, LOGOPTIONS, GLOBAL, and RACLIST or GENLIST. If a class such as OPERCMDS is active with a full complement of profiles on only one database, it may be necessary to embark on a major side project to implement it on the other databases. Be particularly wary of default return code 8 classes, such as JESSPOOL, because you'll need to create profiles to permit access before activating the class the first time. The other major area involves control and audit options. PROTECTALL, Enhanced Generic Naming (EGN), password controls, and OPERATIONS Audit (OPERAUDIT) will need to be the same. For each option, you can choose to revert to the least restrictive setting on all databases or you can implement the option where currently inactive.
- **Started tasks:** This step can be quite complicated. You may encounter started tasks with shared IDs, started tasks with the same name performing different functions, and identical started tasks with differences in attributes such as TRUSTED, OPERATIONS, OMVS uids, and permissions across the system images. You may be forced to rename tasks, reassign IDs, or change permissions to avoid conflicts. IPLs may be necessary to implement changes because some started tasks only initialize at system start-up.
- **User profiles:** Ensure USERIDs found on multiple databases have identical profile and segment attributes such as default group, PROTECTED, RESTRICTED, OMVS uid, and Customer Information Control System (CICS) OPIDENT. Be mindful of products or user interfaces that rely on information in the installation data field such as ViewDirect (a.k.a. INFOPAC) Recipient IDs. If you're forced to change OMVS uids on one database, you may need to make corresponding changes to HFS permissions.
- **Group profiles:** Like user profiles, identically named groups will need consistent attributes. These include Superior Group, Installation Data, TERMUACC, and OMVS gid. If a group with the same name is found in multiple databases but used for different purposes, for instance to grant different sets of users access to dissimilar resources, it will be necessary to implement a replacement group on one or more of the databases.
- **User/group collisions:** You may discover a user on one database is defined as a group on one of the other databases and you'll be forced to convert one of them. The conversion process is more complicated if the user/group has data set profiles because RACF won't let you delete an ID until you have first deleted these profiles. When data set profiles exist, you must delete the data set profiles, delete the user or group, create the replacement user or group, and then re-create the data set profiles. If PROTECTALL is active, you may want to temporarily deactivate it during the collision conversion. Use the IRRRID00 Remove ID utility to help find and delete all unnecessary references to the ID being converted.
- **Data set profiles:** When synchronizing data set profiles, the first task is to identify HLQs common to multiple databases and integrate their profiles to form a single identically matching list. As you add profiles from one database to another, you'll be undercutting the existing profiles and may disrupt existing access (see Figure 2). Be sure to copy the access lists of the existing profiles into the new undercutting profiles as you create them. Once you have matching profiles on all databases, equalize their attributes. You'll need to set UACC, WARNING, AUDIT, access permission lists, and all other profile and segment attributes to be the same. To speed up the merge effort, you may need to set UACCs to the highest level found and wait until after the merge to retighten access.
- **General resource profiles:** The task of synchronizing general resource

profiles is largely similar to data set profiles. There are, however, a few key differences. When databases contain grouping class profiles, you need to synchronize the resource members in the profiles as well as the profiles themselves. If resource members have been defined to more than one profile, careful analysis is required to see what access permissions will be granted in the newly combined RACF database. Similarly, when the same RACFVARS variable is defined on multiple databases but used for different purposes, it may be necessary to replace it with a different variable on some of the databases, and you'll need to change all the profiles that incorporate the variable. For some RACFVARS profiles, &RACLNDE, for example, you'll need to integrate the variable text strings. There are a few creative things you can do to avoid having overlapping, interfering profiles with general resources. Profiles in Job Entry Subsystem (JES)-related classes (e.g., WRITER) are prefixed with the name of the JES subsystem. If JES is named JES2 on multiple images, the resulting resource names and associated profiles might collide. Simply changing the names of the JES subsystems will segregate the various sets of resources. When CICS regions supported by different databases use the same set of resource classes (perhaps the default classes TCICSTRN and GCICSTRN), you can separate them by creating new sets of classes with unique names and have the CICS regions reference different classes. One of the more complicated analysis tasks you'll face is synchronizing profiles in the FACILITY class, since these profiles will affect many software functions and products common to all system images. (For more on this, see the article "Security's Multi-Purpose FACILITY Classes in the April/May 2006 issue of *z/Journal*.)

- **Administrative authorities:** As you synchronize profiles on the different databases, you may need to change profile ownership and group hierarchy. If you use group-level authorities (e.g., Group-SPECIAL), these changes could expand or contract the scope of their authority, prompting you to redesign portions of the group structure. Attempts to synchronize permissions to profiles in the FIELD class or the IRR.PASSWORD.RESET profile in the FACILITY class may force you to alter the administrative responsibilities of certain staff to avoid having their scope of authority become too broad. Also, consider possible changes in the scope of authority for users with Class Authorization (CLAUTH) when setting class POSIT values.

As you complete the process of synchronizing each set of overlapping profiles, ensure they stay synchronized until the final merge and implementation. Institute a strict change management procedure and diligently review RACF commands entered each day. For critical classes and profiles, it may be wise to review their synchronization just prior to implementation.

At the conclusion of the synchronization process, you'll have either completed the migration of all the profiles from one RACF database to the other, or you'll be ready for the final merge process using a utility such as IRRUT400.

### Implementation

Your final step is to complete the merge and implement the combined database across all system images. The tasks involved here will be determined by the merge technique you've chosen. If you've simply copied profiles using commands generated by IBM's free DBSYNC utility, all changes will probably have been done well before implementation. Then, all you need to do on implementation day is copy passwords to the new combined database using IBM's free PWDCOPY utility. If you're using IRRUT400 to perform the final merge to combine databases, note that this utility wasn't designed for use in merging dissimilar databases. Immediately after combining the databases, you'll need to perform additional steps to correct database integrity errors that will emerge. Examples of such errors are broken group connects and dropped access list entries.

Here are some general guidelines to follow in implementing the merged database:

- Execute the final merge steps and implement the shared database during a system maintenance period dedicated to the merge effort while all other system activity is suspended
- Before proceeding, confirm you can log onto and execute RACF commands at the system console so you can enter commands to change RACF options or profiles to recover from mistakes that might prevent Time Sharing Option (TSO) or JES from initializing
- Ensure IRRUT200 backups are prepared and accessible
- Have RVARY command passwords readily available
- Test, test, test
- Be on-call and standing by for rapid problem resolution through at least the first full normal workday and production batch cycle.

Once your z/OS images are running with the new merged database, ensure all performance-enhancing options have been implemented so no system is delayed waiting for RACF to service another. At a minimum, make full, effective use of the Global Access Table, RACLIST, and RACGLIST classes. (For more on this, see the article "10 Ways to Improve RACF Performance" in the October/November 2006 issue of *z/Journal*.)

### Conclusion

The process of building a merged, shared RACF database isn't simple; it requires painstaking analysis. Synchronization of systems and their databases will often require substantial adjustments, which must be implemented with great care. Nonetheless, any RACF administrator who has progressed from managing separate databases to maintaining a single merged one will tell you that the effort was worthwhile. Your management also will be pleased to see the savings in staff resources resulting from the reduction in administration and database maintenance. Finally, your auditors will appreciate the control consistency and ease of auditing inherent with a single RACF database. This is definitely a win-win project.

### About the Authors

**ROBERT S. HANSEL** is founder and president of RSH Consulting, Inc., a firm specializing in RACF. He has worked with RACF for more than 20 years in the roles of manager, administrator, auditor, technician, instructor, and consultant. He's a frequent speaker on numerous RACF topics and supports several RACF users groups.
Email: R.Hansel@rshconsulting.com
Website: www.rshconsulting.com

**ROBERT L. WHITTLE** is a senior RACF consultant at RSH Consulting, Inc. He has been a RACF systems programmer and consultant for more than 15 years. He is the lead developer of RSH Software, Inc.'s products RSH-RDELTA, which compares RACF database profiles, and RSH-RCAPTURE, which reports and compares z/OS and RACF configuration options and tables.
Email: R.Whittle@rshconsulting.com